Preprint of: **A. Reinhardt**, F. Englert, and D. Christin. "Averting the Privacy Risks of Smart Metering by Local Data Preprocessing." In: *Pervasive and Mobile Computing (PMC)* 16 (2015), pp. 171–183.

Averting the Privacy Risks of Smart Metering by Local Data Preprocessing

Andreas Reinhardt[a,*], Frank Englert[b], Delphine Christin[c]

[a]*The University of New South Wales, Sydney, Australia*
[b]*Technische Universität Darmstadt, Darmstadt, Germany*
[c]*University of Bonn, Bonn, Germany*

Abstract

More and more renewable sources are integrated into electric power grids worldwide. Their high generation dynamics, however, require power grid operators to monitor electricity generation and demand at a fine temporal resolution. Even small mismatches between supply and demand can impact the power grid's stability, and thus ultimately lead to blackouts. As a result, smart metering equipment has been widely deployed to collect real-time information about the current grid load and forward it to utilities in a timely manner. Numerous research has shown that power consumption data can, however, reveal the nature of used appliances and their mode of operation at high accuracy. This effectively puts user privacy at risk. In this manuscript, we investigate to which extent the local preprocessing of power data can mitigate this risk. We thus compare the efficacy of different preprocessing steps to eliminate characteristic consumption patterns from the data. Our evaluation shows that a combination of these preprocessing steps can provide a balanced trade-off that is in the interests of both users (privacy protection) and utilities (accurate and timely reporting).

Keywords: Power metering, Privacy protection, Data preprocessing

1. Introduction

One of the key elements of future smart power grids is their integration of renewable sources [1]. The volatile nature of renewables, however, introduces previously unseen uncertainties in the electricity generation. Utility companies hence need to constantly maintain up-to-date knowledge about generation and load in order to avert the risk of power outages. Smart electricity meters have been deployed to this end in many countries [2], as they enable to capture both the distributed generation and the demand of a dwelling. While of immediate benefit to the utilities, the transmission of precise information about the current electric activity in households is often perceived

*Corresponding author. School of Computer Science and Engineering, The University of New South Wales, UNSW Sydney NSW 2052, Australia. Phone +61 2 9385 7679; fax +61 2 9385 5995.

Email addresses: andreasr@cse.unsw.edu.au (Andreas Reinhardt), frank.englert@kom.tu-darmstadt.de (Frank Englert), christin@cs.uni-bonn.de (Delphine Christin)

as a threat to user privacy. This concern is underpinned by research results that have shown that information about the current user activities and even the television content can be inferred based solely on smart meter data (e.g., [3, 4]). So while users may be reluctant to provide high-resolution data because of the possible privacy implications, utilities require exactly this consumption data at a fine temporal resolution in order to adapt the power generation of their non-renewable plants to the dynamically changing demand.

A common way to encounter this problem without forming a trust relationship between customer and utility is the removal of typical characteristics from the data before their transmission. This technique, called *privacy-aware data preprocessing*, has received significant attention in orthogonal domains like participatory sensing [5, 6]. However, the applicability of mechanisms from these domains is very limited due to the different nature of the data collected by smart meters (e.g., the absence of location information). Nonetheless, local preprocessing of sensed data represents a promising way to protect users from potential breaches to their privacy when their consumption data is received by untrusted third parties. In this manuscript we hence investigate to which extent the local preprocessing of power readings can eliminate possibilities to infer appliance types based on their consumption data. To this end, we apply different mechanisms to obfuscate the data and subsequently analyze to which degree appliance types can still be identified after this preprocessing step. The analyzed preprocessing algorithms solely rely on the reporting of slightly altered power consumption readings and do not leverage additional means (e.g., storage batteries [7] or controllable local renewable generation [8]) to physically alter the power demand. It is hence still possible to infer that electrical appliances are operating based on the reported consumption readings. However, when successfully applied, data preprocessing will make it impossible to determine the actual type of an operating appliance or its mode of operation.

Instead of analyzing data that aggregates a complete household's consumption, we focus on distributed smart metering in this manuscript. In this scenario, individual metering devices (sometimes referred to as *smart plugs*) are installed between each appliance's mains plug and the wall outlet. The reasons for selecting this application scenario are twofold. Firstly, existing approaches to infer device activity from smart meter data have shown that the disaggregation of loads performs significantly better when less appliances are connected at the same time [9]. A more efficient privacy protection is thus needed when less appliances are being monitored simultaneously. Secondly, very few household-wide meter data sets (like REDD [10] or Smart* [11]) are freely available. Moreover, these existing data sets are generally neither annotated by the actual appliance activity in the underlying building nor accompanied by the implementation of a disaggregation system. As a result, the effects of local data preprocessing on these data sets cannot be easily determined. In contrast, the Tracebase data set [12] used in this paper contains more than 1,500 appliance power consumption traces, and in combination with our previously presented appliance identification system [12] allows for a better generalization of our results.

This manuscript significantly extends our prior publication [13] by analyzing twice as many preprocessors over larger parameter ranges and assessing the introduced errors in a much more detailed manner. It is structured as follows. First, we provide an overview of related work from the domains of data privacy and smart metering in Sec. 2. Subsequently, we describe our designed software framework and the preprocessing steps in more detail in Sec. 3. Our evaluation settings are explained in Sec. 4, followed by the presentation and discussion of our evaluation results in

Sec. 5. Finally, we conclude this paper in Sec. 6.

2. Related Work

The rise of smart meters has led to the availability of energy consumption readings at an unprecedented time and amplitude resolution. To date, two major applications have emerged that rely on these data. Firstly, knowledge about past, current, and expected energy consumption is vital for the smart grid [1], as it allows utilities to take action in order to maintain the grid's stability. Secondly, value-added services can be based on energy consumption data and cater for the creation of smart buildings [14, 15].

While smart building functionalities can be realized when accurate measurements are available (cf. [16–19]), the same methods can be applied by third parties (e.g., the utility or external attackers) to infer the current situation in a building. Many institutions like the CEN-CENELEC-ETSI Smart Grid Coordination Group, the National Institute of Standards and Technology, or the German Federal Office for Information Security (BSI) have thus defined information security requirements to the smart grid in [20], [21], and [22], respectively. Likewise, many researchers have proposed the use of cryptographic means to ensure a secure transport of data between end users and utilities (e.g., [23–25]). Although proposing a separation of personal information and actual power consumption data, countermeasures to prevent inferring user-specific information from meter data are not described in these documents. Moreover, the generally proposed use of pseudonyms has been shown to be ineffective due to the insufficient number of stakeholders on the electricity market [26].

In order to protect users from such intrusions into their privacy, several solutions have thus been presented in related work. For example, [27] and [28] show how data collected by multiple meters can be aggregated data before sending them to the utility. Similarly, [29] relies on a virtual ring topology, along which meter readings are relayed before being forwarded. While the users are protected against attacks by legitimate receivers of the data (i.e., utilities) in this case, however, they need to trust and cooperate with other household owners. Moreover, transmissions can experience large delays due to the exchanges between clients that precede the final upload to the data recipient, which may render the approaches inapplicable for the highly dynamic nature of smart power grids.

In comparison to collaborative processing approaches, the local privacy-preserving preprocessing of smart meter data has received significantly less attention in the past. Instead of artificially manipulating the collected readings, existing local approaches mainly rely on the use of external energy storage components. The use of batteries to smoothen the load curve and eliminate characteristic features from the data has been presented in [7, 30]. By dynamically adapting the battery output power to a particular appliance's power demand, its existence can be completely hidden. While leading to a potential increase in privacy protection, however, it needs to be remarked that the extent of hiding consumption data this way is inherently limited by the battery capacity. Furthermore, state-of-the-art battery technology suffers from severe limitations, e.g., decreasing capacities over time [31]. Using storage components to protect user privacy may thus not be practical until energy storages become available in large numbers, e.g., as a result of electromobility [32].

Many local processing approaches from orthogonal domains can be leveraged for smart metering scenarios. The addition of noise to sensor measurements has been applied in order to obfuscate user behavior [33], although its effect has not yet been analyzed in the domain of smart electricity grids. Also operating on a local basis, the privacy-aware data preprocessing solution presented in [34] recommends the application of filters to eliminate certain characteristics from power data, but their efficacy has not been analyzed in the domain of smart metering either. Amongst the other mechanisms introduced in Sec. 3.2, we thus regard these approaches in our evaluation and assess their applicability to power consumption data.

3. Concept and Software Framework

The primary objective of this work is to evaluate the extent of privacy protection that can be achieved by preprocessing the data collected by distributed smart meters. In order to analyze the efficacy of data preprocessing steps, we first quantify the privacy threat resulting from the unprocessed transmission of power consumption data. To this end, we use an evaluation system that allows for the classification of appliances based on their power consumption data. Subsequently, results based on preprocessed data are compared to this baseline in order to draw reliable conclusions on the degree of additional protection attained by preprocessing.

To establish the baseline detection accuracy, we rely on our previously designed system that is able to detect the type of an appliance based on its electric power consumption data, which we have presented in [12]. The system extracts specific characteristics that uniquely represent each appliance type based on its power consumption behavior. Subsequently, it leverages a machine learning component to store these characteristics and allow for a later retrieval of device types based on the stored features. When the system is supplied with a power consumption trace collected from another device, it extracts the same features from the provided trace, compares them to the previously established model, and returns the device type with most similar characteristics. The objective of the manuscript at hand, namely obfuscating device-specific characteristics in the power consumption data, should thus lead to larger number of false identifications. Hence, we use the fraction of appliances that can no longer be correctly identified as a measure of the efficacy of our data preprocessing.

3.1. Overall System Architecture

Our overall system is composed of distributed metering units that connect between wall outlets and electric appliances, as well as a server on which the data analysis is performed. This architecture is visualized in Fig. 1. Continuous lines indicate mains connections, whereas dashed lines reflect the wireless data transfer between the meters and the server. We employ Plugwise Circle [35] devices to collect the consumption data due to their commercial availability and their approval for electric safety. All metering units return real power consumption data once per second to the server, which records the power consumption traces in its database for their subsequent processing.

The fundament for the contributions of this paper and the main difference to our original appliance classification system is the addition of a preprocessing step (highlighted in the figure). This step is applied to all power consumption data time series prior to any further processing. By

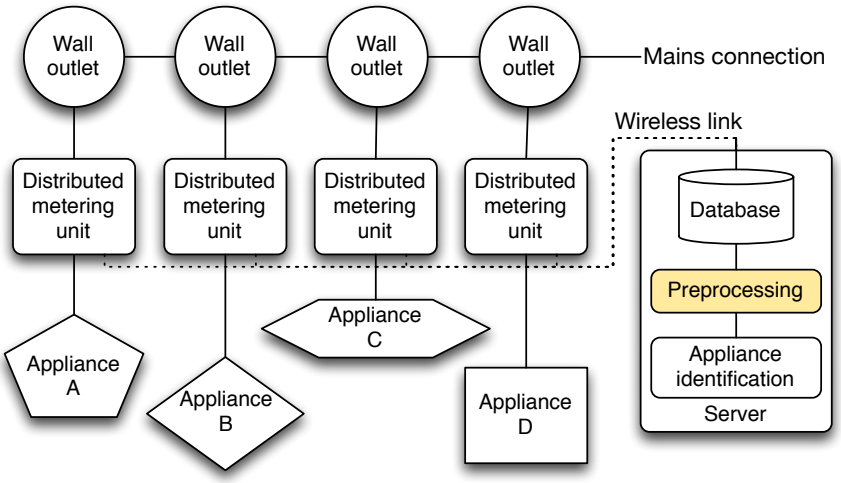


Figure 1: Overall data collection and processing system architecture

preprocessing data locally on the user's premises, potentially compromising characteristics can be removed before the the data is released to third parties. In this manuscript, we investigate multiple alternatives for the data preprocessing step, which we describe in Sec. 3.2 in more detail. Subsequently, the existing appliance identification system is being used to extract representative features from the data stream and facilitate the classification of incoming data streams. We summarize the operation of the appliance classification system, describe the used feature types, and provide details about the machine learning algorithm in Sec. 3.3.

3.2. Data Preprocessing

In the remainder of this paper, we analyze to which degree data preprocessing can help in protecting user privacy. In order to be applicable to the scenario at hand, potential processing algorithm candidates need to fulfill the following two criteria:

1. The algorithms must be sufficiently lightweight to be run on embedded systems like distributed power meters or smart metering infrastructure.
2. The output data of an algorithm needs to retain the general shape of the power consumption curve, i.e., have small deviations from the original data. Similarly, the introduction of excessive time delays between power measurements and their reporting can be expected to hamper grid operations and should thus be avoided.

We have thus selected a set of six data preprocessing filters, which we explain in more detail in the following subsections. To quantify the impact on the privacy protection when temporal dependencies are being considered, we analyze three stateful and three stateless algorithms. The stateful filters take previously observed data into account for the computation of a new output value and are thus also referred to as *time-based* approaches throughout this manuscript. In contrast, the stateless filters modify the signal amplitude independently of any previously observed data, and are called *amplitude-based* filters from here on. In order to highlight the effect of the analyzed

Preprint of: **A. Reinhardt**, F. Englert, and D. Christin. "Averting the Privacy Risks of Smart Metering by Local Data Preprocessing." In: *Pervasive and Mobile Computing (PMC)* 16 (2015), pp. 171–183.

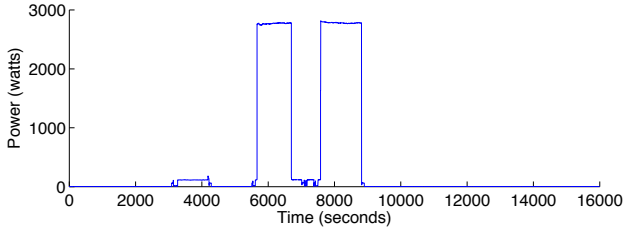
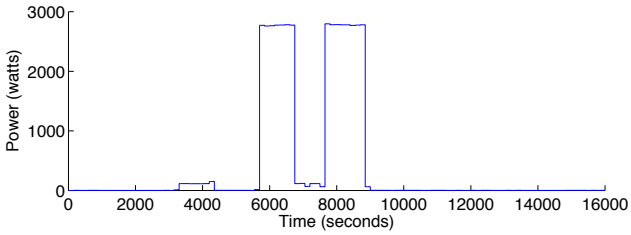
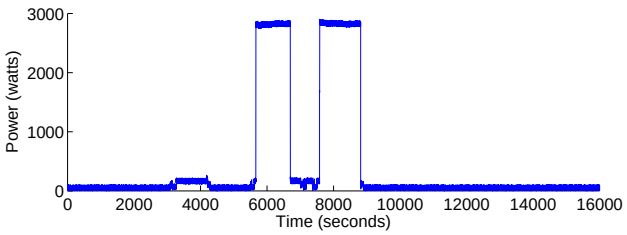


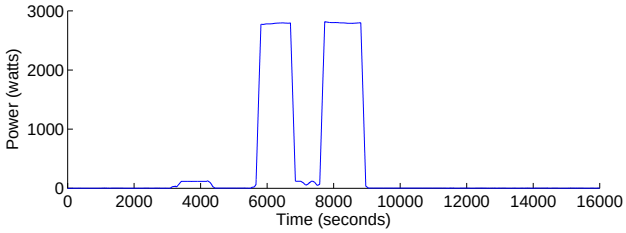
Figure 2: Unprocessed power trace excerpt of a dishwasher



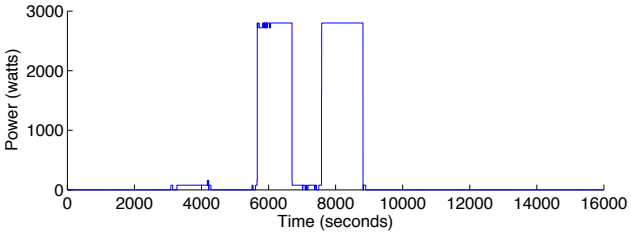
(a) Temporal down-sampling with a window size of w = 150 s



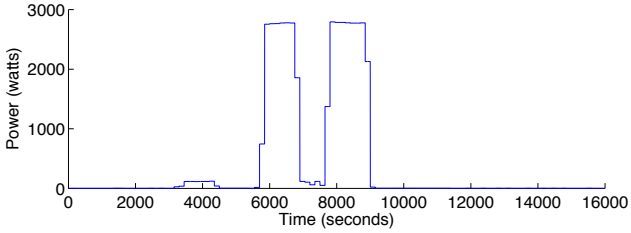
(b) Addition of Gaussian noise with an amplitude of a = 100 W



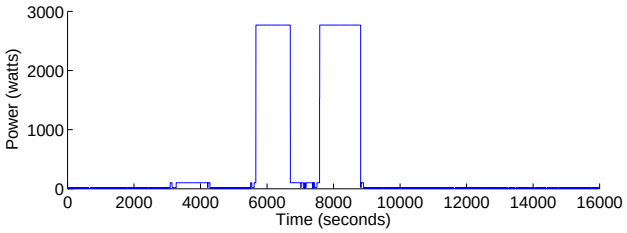
(c) Temporal averaging with a window size of w = 150 s



(d) Quantization with a factor q = 80 W



(e) Averaged down-sampling with a window size of w = 150 s



(f) Clustered quantization with n = 22 output clusters

Figure 3: Comparison of the resulting traces when the six preprocessing filters are applied to a dishwasher's power consumption trace

processing algorithms on real-world data, we visualize their impacts on an excerpt from a dishwasher's operation cycle. The unaltered consumption trace is depicted in Fig. 2 for reference.

3.2.1. Temporal Down-Sampling

Temporal down-sampling is a mechanism to intentionally reduce the temporal fidelity of power consumption readings. To this end, it periodically takes a sample of the data and reports the same

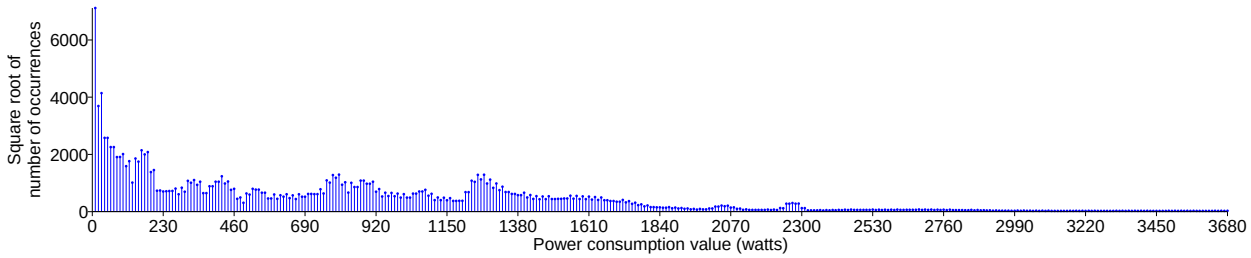


Figure 4: Histogram of the entire used input data (the y-axis shows the square root of the actual value for improved visual clarity)

power consumption value for a time duration w. Should new samples be received during w, these values are discarded and the previous value is repeated instead. While down-sampling thus maintains the temporal frequency at which measurements are made available (e.g., one sample per second in case of the Plugwise devices introduced above), in the worst case a potential change of the actual power consumption will only be reported after a complete window w has passed. Furthermore, as sensor readings are intentionally discarded, the error introduced by the application of temporal down-sampling is only limited by maximum power measurement capability of the underlying sensing device, but unbounded in theory. Fig. 3a shows the output of the down-sampling step when a down-sampling interval of w=150 seconds is being used.

3.2.2. Temporal Averaging

Instead of omitting input data samples from their forwarding to the processing system, temporal averaging takes all collected readings into consideration and can thus be expected to lead to smaller deviations between actual and reported data. We have used a window-based averaging function that reports the arithmetic mean of the previous w sensor readings. The output of our averaging preprocessor for w=150 seconds is shown in Fig. 3c, clearly showing the smoothened consumption pattern. Averaging introduces a time lag, and may thus only be applicable in scenarios where this delay can be tolerated by the recipient of the data.

3.2.3. Temporal Averaging and Down-Sampling

This preprocessing alternative is the combination of an averaging of the input data and the down-sampling of the resulting value as described in the previous two paragraphs. Again, the mean value of the previously collected sensor readings is continuously calculated over a sliding time window of w seconds. The resulting value is, however, down-sampled and only reported once at the beginning of every window and then repeated until w has passed. The output of this approach for w=150 seconds is shown in Fig. 3e, from which the characteristic steps on the steep edges of the power consumption curve become apparent. Like the previously described averaging step, a time lag is introduced when using this approach.

3.2.4. Noise Addition

Adding noise to the signal is a stateless way of modifying power consumption readings. Characteristic fluctuations of an appliance's power consumption that only have small amplitudes can

be covered in the added noise, thus potentially leading to a higher privacy protection. In this preprocessor, we have used a noise source that returns uniformly distributed values with an amplitude between $-a$ and $+a$, which are added to incoming sensor data. The result of adding noise with a=100 W to the dishwasher's consumption trace is shown in Fig. 3b. As no negative power consumption values should result from the addition of noise, this preprocessor has been configured to return the absolute value of the computation result. In order to eliminate dependencies on the random number generator's seed, we have run each of the evaluation experiment multiple times with different seeds and only show the resulting mean values.

3.2.5. Linear Value Quantization

Value quantization is realized by rounding the actual power consumption values to a multiple of a pre-defined quantization factor q. Because the quantization step is stateless and requires no historical data, no delay is introduced by the introduction of this preprocessing step. The application of quantization to the dishwasher's consumption data is shown in Fig. 3d for q=80 W. It can be seen that quantization eliminates the slight slope on top of the power-intensive heating periods while the general shape is maintained. The decision to utilize a quantization step has been supported by the fact that many electricity meters implicitly quantize values by outputting a number of pulses for each consumed unit of energy (e.g., 1,000 pulses per kWh of consumed energy).

3.2.6. Adaptive Cluster-based Quantization

In contrast to defining clusters of equally sized value ranges, this preprocessing step adapts to the actual characteristics of the input data. To this end, we have computed the histogram of all input data (cf. Sec. 4), which is visualized in Fig. 4. In order to intentionally create ambiguities between similar power consumption readings to help protecting user privacy, we have applied the Mean Shift algorithm [36] to separate the histogram into clusters. For all input data the falls within the boundaries of any of the resulting clusters, the median value of the corresponding cluster's elements is reported. The algorithm can be parameterized to either return more clusters and thus less errors between unprocessed and processed value, or to use less clusters and introduce larger errors. We show the dishwasher's consumption trace when clustering the histogram into n=22 individual sections in Fig. 3f.

3.3. Classification and Features

The evaluation of the achievable privacy protection is based on our appliance classification framework presented in [12], which follows the overall process flow shown in Fig. 5. Let us briefly revisit its operation. At first, power consumption traces of 24 hours duration are collected from electric appliances. From a subset of the traces, characteristic features are extracted, annotated by the type of the underlying appliance, and stored in the form of feature vectors. Each of the resulting annotated feature vectors is subsequently forwarded to the machine learning component, where a classification model is constructed based on the annotated data. This phase during which the model is constructed is termed the *training phase* (cf. the upper part of Fig. 5). Subsequently, the remaining traces are inserted into the system, their feature vectors are computed

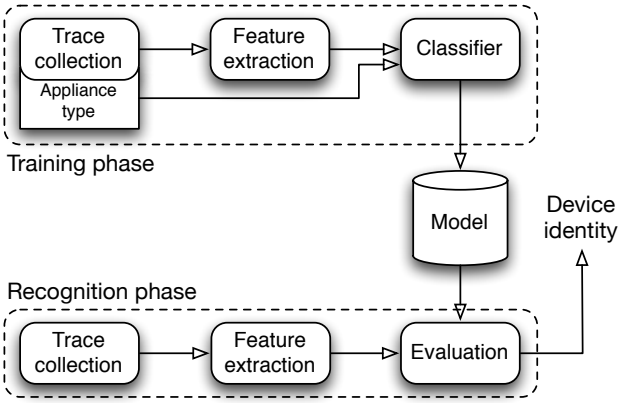


Figure 5: Appliance classification architecture

without class annotations, and the classifier's output is compared to the actual device class (*recognition phase*). We have used a 25-fold cross-validation approach to evaluate the accuracy of the established model, i.e., 96% of the input traces were used to train a model, against which the remaining 4% of the traces were subsequently evaluated. This process is repeated for all 25 possible permutations of training and recognition data and the average accuracy values across all combinations are reported. A supplementary analysis of the impact of the actual random seed that is used to split the input data into training and testing sets has indicated that only minimal deviations can be observed as a result of choosing different seeds. Hence, all results reported in the rest of this manuscript are based on a single run of the 25-fold cross validation.

Similar to [17] and [37], our system utilizes more than 500 different features from different domains in order to describe the characteristic properties of the power consumption traces. We regard features from both the temporal and frequency domain in order to incorporate both the sudden changes encountered on appliance activation as well as periodicities throughout the day into our classification model. More details on the employed four classes of features are provided in the following subsections.

3.3.1. Temporal Appliance Behavior

This class of features encompasses information about the typical operation hours of a device as well as the days of the week that it is being operated. Furthermore, information about the number of activity cycles per day and their minimum, average, and maximum lengths are recorded. Separate features model whether the duration of active phases varies throughout the day and what typical ranges of these variations are.

3.3.2. Energy and Power Consumption Levels

The energy and power consumption characteristics are also extracted for different periods throughout the day. Both minimal and maximal values are considered, as well as averaged consumption values throughout activity periods. Besides calculating typical activity operation power levels, the variance and possible ascending or descending trends of their length and consumption

throughout the day are captured. The distribution of the observed power readings between their minimal and maximal value is considered as well. Finally, appliances that only draw a constant vampire power (e.g., Internet routers) are identified.

3.3.3. Shape of the Power Consumption

Taking both time and power consumption into account at the same time, this third class of features regards the actual shape of the daily power consumption curve. This includes both the steepness of initial inrush currents observed upon device activation as well as the characteristic oscillations in steady-state operation. We apply different thresholds (e.g., 50, 200, and 2,000 watts) and count the number of times each of these thresholds is crossed as well as how long the appliance operates in each segment. We also specifically consider the shape of the highest peak per activity period in terms of its slopes and overshoots.

3.3.4. Statistic Features

In addition to regarding the power consumption during short segments, e.g., initial peaks or sudden changes in the consumption, we also consider the statistics of the complete diurnal trace by calculating its spectrum. The most dominant frequency coefficients are then used as features in the appliance classification. We also compare different activity intervals to each other by means of calculating their cross-correlations in order to determine the characteristics of periodically occurring operational cycles. Finally, we calculate a histogram over the power consumption ranges throughout a day and analyze the distribution of the appliance's power consumption across each compartment.

3.4. Achievable Classification Accuracy

Our previous results have shown that classification accuracy values in excess of 90% can be achieved when all of the presented features are being used for the appliance classification [12]. In other words, a very large fraction of the input data (composed of more than a thousand appliance traces) could be correctly classified solely based on their power consumption data throughout a 24-hour period. In our evaluations we have demonstrated that maximum and average power consumption values are the most important features for the classification of appliances. Based on this observation, we have specifically chosen to preprocess the power consumption data in ways that alter the consumption characteristics and analyze their impact on the classification accuracy. While our previous work has thus effectively promoted *anti-privacy* by identifying the types of electric appliances, we address the opposite target in this manuscript, namely how data preprocessing can render our appliance identification system ineffective.

4. Evaluation Setup

Our evaluation is based on the software system presented in Sec. 3. We have installed the server components on a dedicated machine that runs the database, the preprocessing modules, and the appliance identification engine. For the construction of the classification model, we have used the Weka data mining toolkit [38]. Based on the comparison of different classifiers in our previous work, we have chosen to use the Random Forest classifier for the machine learning step, as it has

Table 1: Power consumption traces used in the evaluation

Device type	# appliances	# traces
Alarm clock	1	5
Bean-to-cup coffee maker	1	43
Bread cutter	1	12
Coffee maker	5	77
Cooking stove	1	16
Desktop computer	9	126
Dishwasher	3	65
Ethernet switch	3	11
Freezer	1	9
HDTV media center	1	5
HiFi stereo amplifier	3	88
Internet router	1	20
Iron	1	3
Lamp	6	77
Laptop computer	6	50
Microwave oven	5	51
Monitor (CRT)	2	14
Monitor (TFT)	14	178
Playstation 3 console	2	12
Powered USB hub	1	10
Printer	1	6
Projector	1	8
Refrigerator	8	189
Solar-thermal system	1	8
Subwoofer	2	28
Television set	10	138
Toaster	4	21
Tumble dryer	2	9
USB hard disk drive	4	29
Vacuum cleaner	1	1
Video projector	1	19
Washing machine	7	50
Water fountain	1	56
Water kettle	8	115
Xmas lights	1	6
Total	119	1,555

resulted in the highest classification accuracy for the task at hand [12] and has a fast execution time.

The data for the classification has been taken from our Tracebase project [12]. The Tracebase already features more than 1,200 diurnal power consumption traces of more than 30 household appliance types. Furthermore, we have collected more than 300 additional traces in order to base our evaluation on an even larger corpus of data. On average, the power consumption traces have been collected at a high granularity of one sample per second and with an amplitude resolution of one watt. In total, we have used 1,555 power consumption traces collected from 35 different appliance types in our evaluation, as listed in Table 1.

In order to put the achieved device classification results into perspective, we compare them to the baseline, in which no preprocessing steps are applied (i.e., the parameters are chosen as q=1 watt, w=1 second, no added noise). Subsequently, we conduct a comprehensive analysis of the classification accuracy when varying the parameter values across a large range of values. More precisely, we have chosen the following boundaries for the parameter ranges:

- For the window size w, 45 discrete values in the range from 1 to 850 seconds have been

analyzed. The same window size has been used for both averaging and down-sampling as well as the preprocessor that combines both filters.

- Both the quantization factors q and the noise power amplitude a have been varied from 1 to 180 watts in 30 discrete steps.
- In the cluster-based quantization approach, the Mean Shift algorithm's threshold parameter has been varied in order to obtain a different number of clusters. In our evaluation, we have used four different settings of the parameter (25, 50, 100, and 200 watts). As a result, the algorithm returned 67, 22, 10, and 6 clusters, respectively, based on the histogram of all input data as shown in Fig. 4.

5. Evaluation

In this section, we conduct a comprehensive evaluation of the parameter space in order to quantify the improvements to user privacy protection offered by data preprocessing. In a first experiment, we determine the bounds for the classification success rates in order to put all further results into perspective. Subsequently, we analyze the impact of the preprocessing steps on the appliance classification accuracy that serves as our privacy preservation metric. In a supplementary simulation study, we furthermore quantify the error that is added to the data by the preprocessing algorithms and weigh it against the requirements of utility companies. We conclude this section with a summary of the observed results.

5.1. Baseline Classification Performance

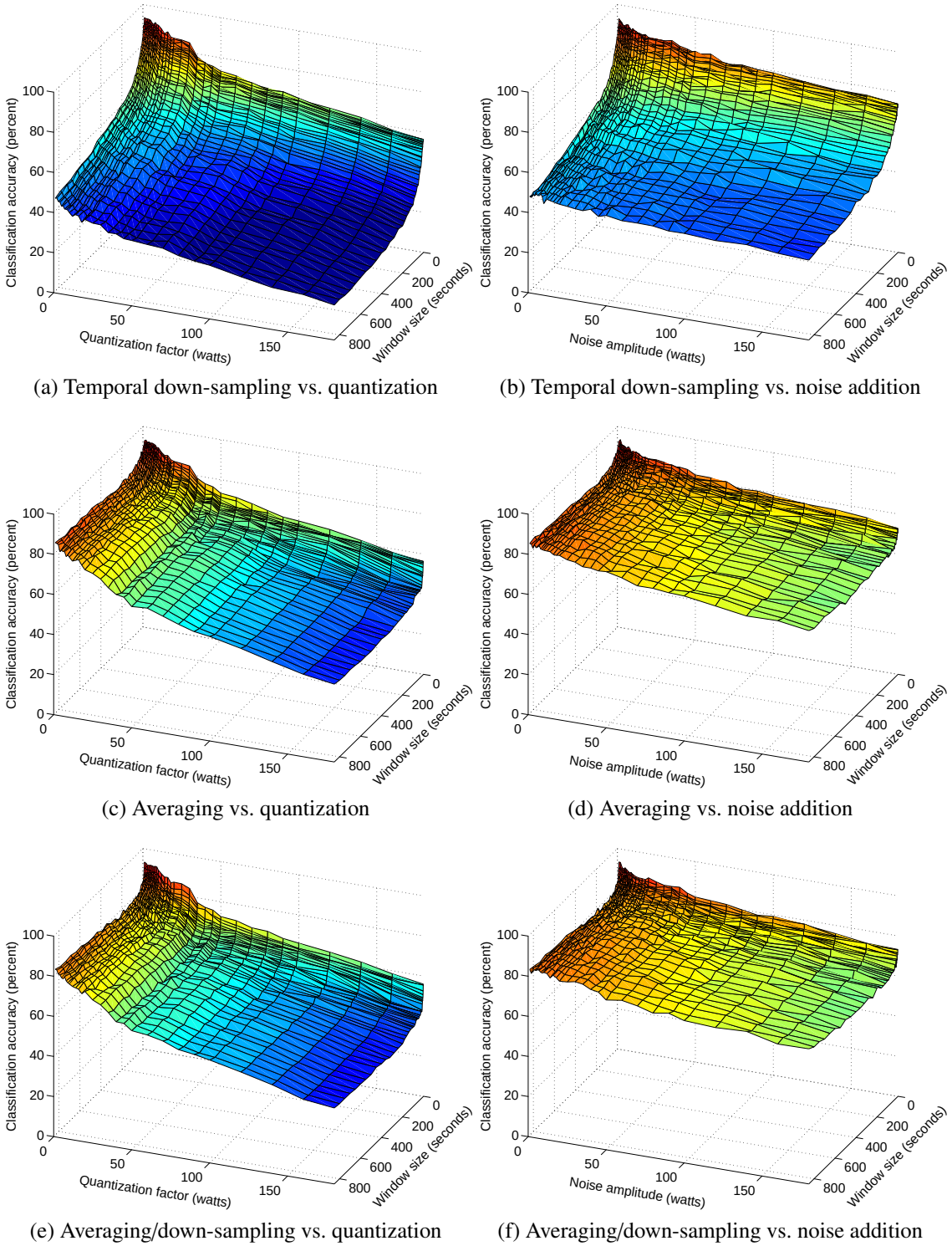
In order to put the evaluation results into perspective, we have first evaluated the baseline detection accuracy for all 1,555 input data traces. In this case, the application identification component has returned an achievable accuracy value of 90.5%, i.e., nine out of ten devices could be correctly identified solely based on a 24-hour long sample of their power consumption. Likewise, the worst classification result is equal to randomly guessing an appliance's type, and can thus be calculated as $1/\#appliances$. For the given input set of 35 appliance types, the minimum accuracy thus equals 2.9%.

5.2. Classification Accuracy

First, we analyze to which extent the preprocessing filters exert an impact on the classification accuracy. To this end, we have computed the accuracy values for all possible combinations of the three time-based preprocessors (down-sampling, averaging, and the combination of both) with the three amplitude processors (linear quantization, noise addition, and quantization based on the histogram of all input data). Please note at this point that (as shown in Fig. 1) all input traces are preprocessed before they are being supplied to the appliance classification component in order to cater for a fair evaluation.

The classification accuracy values when quantization and noise addition are combined with the temporal preprocessors are shown in Fig. 6. As expected, when the quantization factor (or noise amplitude) is set to 1 W and a window size of 1 s is chosen, the reference accuracy of 90.4% is reached in all plots. The impact of time-based preprocessors can be seen on the left 2D plane where

Preprint of: **A. Reinhardt**, F. Englert, and D. Christin. "Averting the Privacy Risks of Smart Metering by Local Data Preprocessing." In: *Pervasive and Mobile Computing (PMC)* 16 (2015), pp. 171–183.



(a) Temporal down-sampling vs. quantization

(b) Temporal down-sampling vs. noise addition

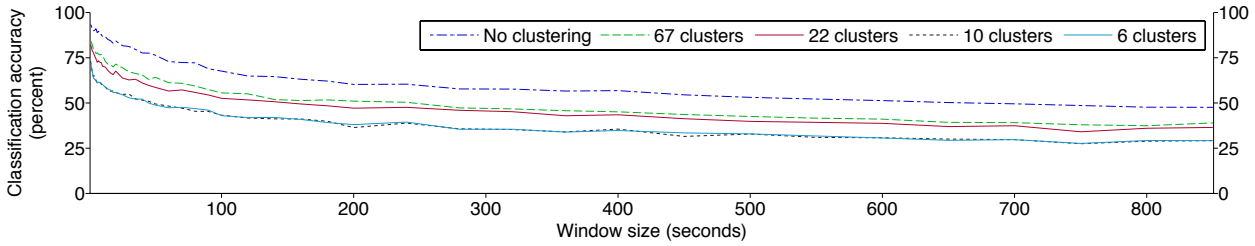
(c) Averaging vs. quantization

(d) Averaging vs. noise addition

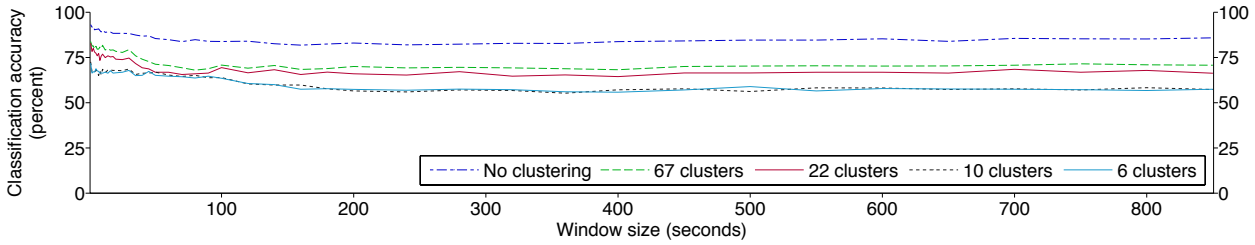
(e) Averaging/down-sampling vs. quantization

(f) Averaging/down-sampling vs. noise addition

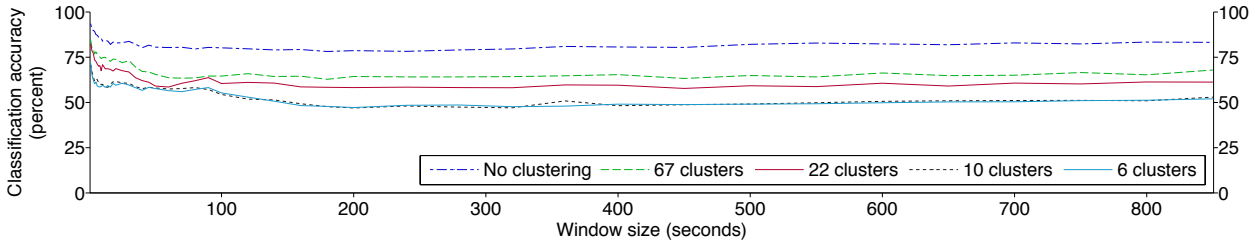
Figure 6: Resulting classification accuracies when the analyzed preprocessing filters have been applied to all traces in the input set



(a) Temporal down-sampling vs. cluster-based quantization



(b) Averaging vs. cluster-based quantization



(c) Averaging/down-sampling vs. cluster-based quantization

Figure 7: Resulting classification accuracies when the cluster-based quantization step has been applied to all traces in the input set

q=1 W. Likewise, the impact of preprocessors that change the amplitude of the signal becomes apparent on the back plane, i.e., w=1 s. This analysis of each time-based preprocessor's individual impact already makes clear that down-sampling leads to a much lower classification accuracy (and thus a higher privacy protection) than averaging or their combination. In fact, even the largest analyzed temporal window size still leads to a correct classification rate of 85.9% for averaging, and 83.5% for the combination of averaging and down-sampling. In contrast, down-sampling already reaches this classification accuracy at a window size of only 20 s, and leads to a rate of only 47.1% correct classifications when the largest window size is chosen.

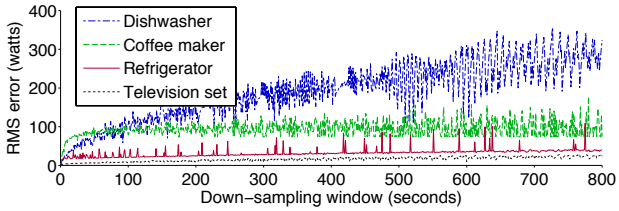
With regard to the impact of the amplitude-based preprocessors (i.e., the curve's behavior on the 2D plane in the back of the diagrams), it becomes clear that both quantization and the addition of noise lead to similar results up to window sizes of 30 W. When linear quantization is being used, however, the classification accuracy experiences a measurable drop above this quantization factor, eventually leading to a classification accuracy of 56.0% for q=180 W. In contrast, 71.5% of all traces are still being correctly classified after the addition of noise with an amplitude of a=180 W. The nature of the Mean Shift clustering algorithms disallows for the specification of the number of clusters, but instead relies on defining the allowed bandwidth, i.e., the allowed distance of a

new data point to an existing cluster. Hence, Fig. 7 shows the resulting classification accuracies when the previously introduced threshold parameters are being used. Again ignoring the impact of the time-based preprocessors (i.e., at a value of w=1 s on the x-axis), the plots confirm the 90.4% baseline accuracy when no clustering is applied, which reduces to 71.8% when only 6 clusters are being used. When combined with time-based preprocessors, temporal down-sampling again achieves the largest reductions in classification accuracy. Based on quantization into 6 clusters, its accuracy values reaches 29.9%, as compared to 57.4% for temporal averaging and 52.0% for both combined.

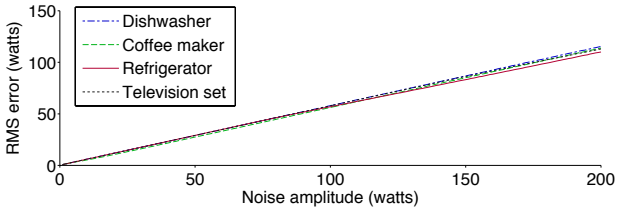
Taking the combined results into consideration, first of all it becomes clear that the combination of adding noise and averaging (cf. Figs. 6d and 6f) leads to very limited privacy protection. This behavior is, however, expected as the underlying concepts of these two preprocessors are diametrically opposed. Furthermore, it can be observed that the relatively highest losses of prediction accuracy are encountered when only small parameter values are being chosen. Across all evaluations, more than half of the maximum reduction of classification accuracy is already achieved when the window is 120 s or larger. In all cases, the application of temporal down-sampling leads to the highest reductions in classification accuracy, whereas the differences between averaging and the combination of down-sampling and averaging are minor. Overall, the highest privacy protection levels are achieved when combining quantization and down-sampling. Setting w=850 s and q=180 W leads to an overall classification accuracy of only 36.9% (see Fig. 6a), and when replacing the linear quantization by the histogram-based clustering with few clusters instead, even larger reductions can be achieved.

5.3. Errors Introduced by Data Preprocessing

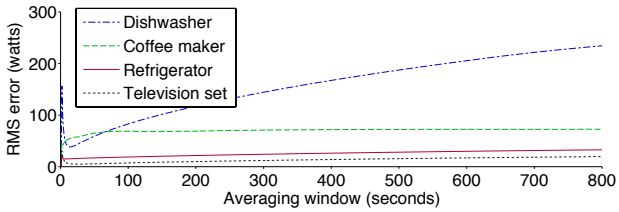
By applying any of the presented preprocessing steps, the input signal is altered from its original form. We hence analyze this introduced error next. To this end, we determine the root-mean-square (RMS) error between the original and the preprocessed power consumption traces, as proposed in [27]. The maximum power consumptions observed by the considered appliances are 2,866 W (dishwasher), 1,488 W (coffee maker), 1,461 W (refrigerator), and 284 W (television set), respectively. As all preprocessing steps that involve temporal down-sampling suffer from unbounded errors, the largest errors are reported for the dishwasher appliance in this case (Figs. 8a and 8e). The error of the averaging step shown in Fig. 8c follows a similar shape, but is slightly lower due to the prior smoothing of the data and the resulting elimination of spikes in an appliance's power demand. In contrast to the time-based preprocessing, the remaining three subgraphs show the effect when only the amplitude of the signal is changed. The expected linear relationship between the added uniformly distributed noise and the RMS error is confirmed in Fig. 8b. Although the quantization error is bounded by $|q/2|$ (i.e., 100 W for a quantization factor q=200 W), the linear quantization mechanism only results in RMS errors of less than 60 W across the four devices (see Fig. 8d). Finally, Fig. 8f shows the errors when different numbers of clusters are extracted from the histogram of all input data. Larger errors are introduced when 20 or less clusters are being used, whilst a larger number of clusters leads to a more approximate representation of the data and thus to smaller errors. As a general observation, it becomes apparent that amplitude-based preprocessors introduce smaller errors than their time-based counterparts.



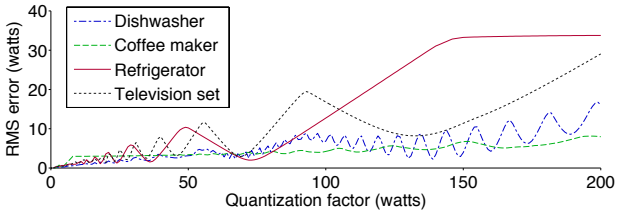
(a) RMS power errors for different down-sampling window sizes



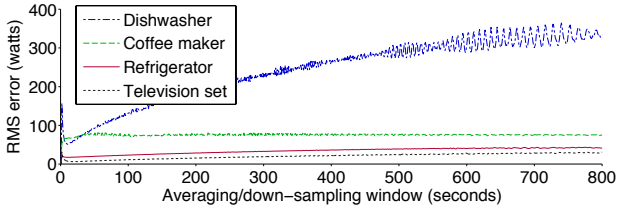
(b) RMS power errors for different noise amplitudes



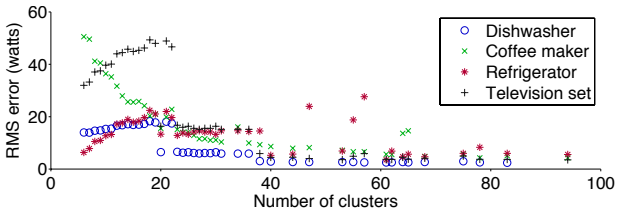
(c) RMS power errors for different averaging window sizes



(d) RMS power errors for different quantization factors



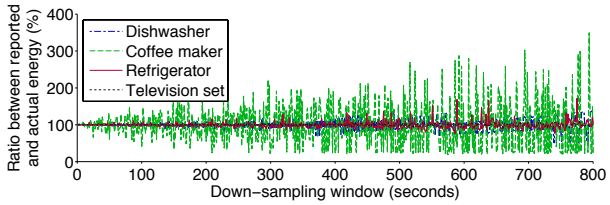
(e) RMS power errors for different averaged down-sampling window sizes



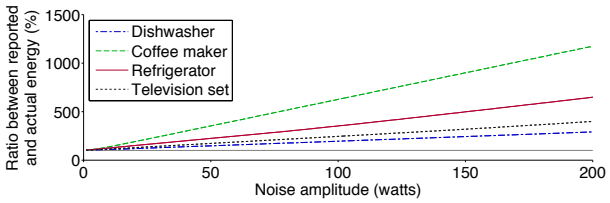
(f) RMS power errors for different numbers of quantization clusters

Figure 8: RMS power errors when the analyzed preprocessing filters have been applied to traces of different appliance types

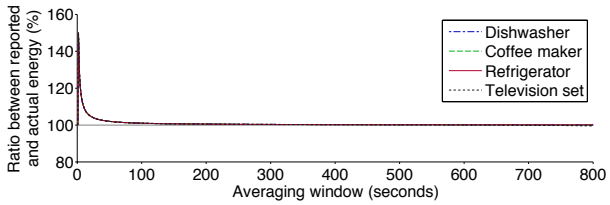
In a final analysis, we consider the differences between an appliance's actual daily energy demand and the reported energy consumption after preprocessing has been applied. The results are visualized in Fig. 9, in which a light gray line also indicates a ratio of 100%, i.e., an exact match between actual and reported daily energy consumption. For reference, the total daily energy demands of the considered traces were 1.20 kWh (dishwasher), 0.22 kWh (coffee maker), 0.38 kWh (refrigerator), and 0.66 kWh (television set). The diagrams show that preprocessing leads to the reporting of a lower energy consumption in some cases (e.g., when applying the quantization step with q>150W to the refrigerator's consumption, as shown in Fig. 9d), but also to situations where a higher energy demand is reported (cf. the noise addition shown in Fig. 9b). Besides the huge introduced errors of up to 1,200% when random noise is added to the signals, it can however be observed that the discrepancies mostly stay within a band of 50% to 200% of the original energy demand. Even when down-sampling with large window sizes is being applied (cf. Fig. 9a), the reported energy consumptions only experience moderate deviations of at most a factor of 3.4 from the ground truth.



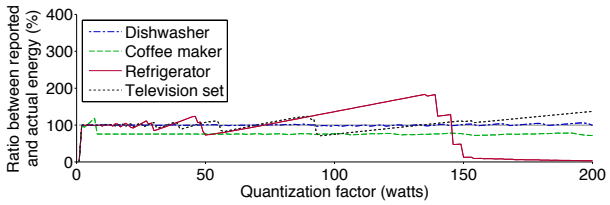
(a) Ratio between reported and actual energy consumption for different down-sampling window sizes



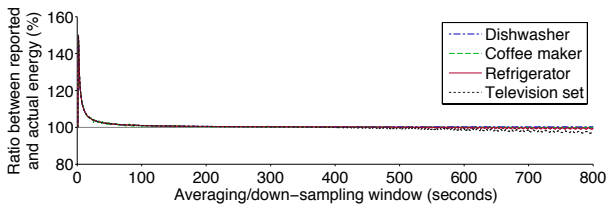
(b) Ratio between reported and actual energy consumption for different noise amplitudes



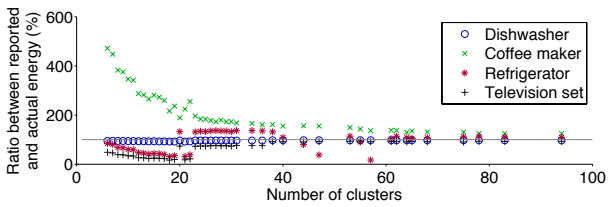
(c) Ratio between reported and actual energy consumption for different averaging window sizes



(d) Ratio between reported and actual energy consumption for different quantization factors



(e) Ratio between reported and actual energy consumption for different averaged down-sampling window sizes



(f) Ratio between reported and actual energy consumption for different numbers of quantization clusters

Figure 9: Ratios between reported and actual energy consumption when the analyzed preprocessing filters have been applied to traces of different appliance types

5.4. Evaluation Summary

From the simulations, it has become clear that all presented approaches are suitable to achieve a reduction of the classification accuracy, i.e., an increase in privacy protection. The efficacy of the algorithms, however, strongly varies. When analyzed individually, both averaging and the combination of averaging and down-sampling have only provided a minimal improvement in privacy protection, but lead to considerable errors being added to the signal. Similarly, the addition of noise has only reduced the classification accuracy by less than 20 percentage points, but introduced measurable errors. In contrast, a reduction by more than 40 percentage points was achieved when temporal down-sampling with large window sizes has been applied. In combination with either version of the quantization filter (linear or clustered), the best privacy protection results were achieved.

The large RMS error of temporal down-sampling and the corresponding deviations in the reported energy demand, however, may render its usage inapplicable for some scenarios. Despite the fact that quantization errors are likely to even out across a large field of participants (e.g., in smart grids) due to the law of large numbers, its application might be less favorable when the

data is, e.g., used for billing purposes; this is especially true when an energy demand below the actual value is being reported. In conclusion, we however still believe that our comprehensive analysis of a range of preprocessors represents a key element to make informed decisions for privacy-preserving preprocessing that can be adapted to any power metering scenario.

6. Conclusions

The protection of user privacy is a key element of today's society. With the rise of smart metering, a novel sensing modality has emerged that can be leveraged to draw fine-grained portraits of the activities in a household. We have thus analyzed how the application of preprocessing algorithms to distributed smart metering data can be used to mitigate these privacy risks. To this end, we have studied the impact of six preprocessing filters and their combinations on more than 1,500 power consumption traces. When any of the proposed preprocessing steps has been applied to the data, the classification accuracy has experienced a degradation, i.e., the privacy protection has increased, however to a variable degree.

Although the highest privacy protection results have only been achieved when significant errors were introduced, the filters can be tuned to provide the desired trade-off between privacy and reporting error. In fact, even small parameter settings can lead to good protection. For example, applying linear quantization with q=45 W and temporal down-sampling with w=90 s already leads to a situation in which only half as many appliances can be identified and the introduced error is below 100 W on average. In summary, our proposed approach has proven that users can increase their privacy protection at the cost of intentionally inaccurate data reporting.

References

[1] International Energy Agency, Technology Roadmap "Smart Grids", available online: `http://www.iea.org/topics/smartgrids/publications/` (2011).

[2] R. Hierzinger, M. Albu, H. van Elburg, A. J. Scott, A. Łazicki, L. Penttinen, F. Puente, H. Sæle, European Smart Metering Landscape Report, Online: `http://www.smartregions.net/` (2012).

[3] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, D. Irwin, Private Memoirs of a Smart Meter, in: Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems For Energy-Efficiency In Buildings (BuildSys), 2010, pp. 61–66.

[4] U. Greveler, B. Justus, D. Loehr, Multimedia Content Identification Through Smart Meter Power Usage Profiles, in: Proceedings of the International Conference on Information and Knowledge Engineering (IKE), 2012, pp. 383–390.

[5] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, N. Triandopoulos, AnonySense: Privacy-aware People-centric Sensing, in: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys), 2008, pp. 211–224.

[6] D. Christin, A. Reinhardt, S. S. Kanhere, M. Hollick, A Survey on Privacy in Mobile Participatory Sensing Applications, Journal of Systems and Software 84 (11) (2011) 1928–1946.

[7] G. Kalogridis, C. Efthymiou, S. Denic, T. Lewis, R. Cepeda, Privacy for Smart Meters: Towards Undetectable Appliance Load Signatures, in: Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm), 2010, pp. 232–237.

[8] A. Alarcon-Rodriguez, G. Ault, S. Galloway, Multi-Objective Planning of Distributed Energy Resources: A Review of the State-of-the-Art, Renewable and Sustainable Energy Reviews 14 (5) (2010) 1353–1366.

[9] A. Marchiori, D. Hakkarinen, Q. Han, L. Earle, Circuit-Level Load Monitoring for Household Energy Management, IEEE Pervasive Computing 10 (1) (2011) 40–48.

[10] J. Zico Kolter and Matthew Johnson, REDD: A Public Data Set for Energy Disaggregation Research, in: Proceedings of the SustKDD Workshop on Data Mining Applications in Sustainability (SustKDD), 2011, pp. 1–6.
[11] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, J. Albrecht, Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes, in: Proceedings of the Workshop on Data Mining Applications in Sustainability (SustKDD), 2012, pp. 1–6.
[12] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, R. Steinmetz, On the Accuracy of Appliance Identification Based on Distributed Load Metering Data, in: Proceedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT), 2012, pp. 1–9.
[13] A. Reinhardt, F. Englert, D. Christin, Enhancing User Privacy by Preprocessing Distributed Smart Meter Data, in: Proceedings of the 3rd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT), 2013, pp. 1–7.
[14] J. Granderson, M. A. Piette, B. Rosenblum, L. Hu, Energy Information Handbook: Applications for Energy-Efficient Building Operations, Lawrence Berkeley National Laboratory LBNL-5272E (2011).
[15] T. A. Nguyen, M. Aiello, Energy Intelligent Buildings based on User Activity: A Survey, Energy and Buildings 56 (2013) 244–257.
[16] M. Berges, E. Goldman, H. S. Matthews, L. Soibelman, Enhancing Electricity Audits in Residential Buildings with Nonintrusive Load Monitoring, Journal of Industrial Ecology 5 (14) (2008) 844–858.
[17] J. Liang, S. K. K. Ng, G. Kendall, J. W. M. Cheng, Load Signature Study – Part I: Basic Concept, Structure, and Methodology, IEEE Transactions on Power Delivery 25 (2) (2010) 551–560.
[18] M. Kazandjieva, O. Gnawali, B. Heller, P. Levis, C. Kozyrakis, Identifying Energy Waste through Dense Power Sensing and Utilization Monitoring, Tech. Rep. CSTR 2010-03, Stanford University (2010).
[19] C. Beckel, L. Sadamori, S. Santini, Automatic Socio-Economic Classification of Households Using Electricity Consumption Data, in: Proceedings of the 4th International Conference on Future Energy Systems (e-Energy), 2013, pp. 75–86.
[20] CEN-CENELEC-ETSI Smart Grid Coordination Group, Smart Grid Information Security, Online: `http://ec.europa.eu/energy/gas_electricity/smartgrids/doc/xpert_group1_security.pdf` (2012).
[21] The Smart Grid Interoperability Panel – Cyber Security Working Group, Guidelines for Smart Grid Cyber Security: Vol. 2, Privacy and the Smart Grid, National Institute of Standards and Technology Report NISTIR 7628 (2010).
[22] H. Kreutzmann, S. Vollmer, N. Tekampe, A. Abromeit, Protection Profile for the Gateway of a Smart Metering System (Gateway PP), Protection Profile 01.01.01 (final draft), Federal Office for Information Security Germany (2011).
[23] A. R. Metke, R. L. Ekl, Security Technology for Smart Grid Networks, IEEE Transactions on Smart Grid 1 (1) (2010) 99–107.
[24] F. D. Garcia, B. Jacobs, Privacy-Friendly Energy-Metering via Homomorphic Encryption, in: Proceedings of the 6th International Workshop on Security and Trust Management (STM), 2010, pp. 226–238.
[25] A. Rial, G. Danezis, Privacy-Preserving Smart Metering, in: Proceedings of the Workshop on Privacy in the Electronic Society (WPES), 2011, pp. 49–60.
[26] M. Stegelmann, D. Kesdogan, GridPriv: A Smart Metering Architecture Offering k-Anonymity, in: Proceedings of the 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012, pp. 419–426.
[27] L. Sankar, S. Kar, R. Tandon, H. V. Poor, Competetive Privacy in the Smart Grid: An Information-theoretic Approach, in: Proceedings of the 2nd IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011, pp. 220–225.
[28] K. Kursawe, G. Danezis, M. Kohlweiss, Privacy-Friendly Aggregation for the Smart-Grid, in: Proceedings of the 11th International Conference on Privacy Enhancing Technologies (PETS), 2011, pp. 175–191.
[29] M. Badra, S. Zeadally, Design and Performance Analysis of a Virtual Ring Architecture for Smart Grid Privacy, IEEE Transactions of Information Forensics and Security 9 (2) (2014) 321–329.
[30] W. Yang, N. Li, Y. Qi, W. Qardaji, S. McLaughlin, P. McDaniel, Minimizing Private Data Disclosures in the Smart Grid, in: Proceedings of the ACM Conference on Computer and Communications Security (CCS), 2012,

pp. 415–427.

[31] G. Ning, B. N. Popov, Cycle Life Modeling of Lithium-Ion Batteries, Journal of The Electrochemical Society 151 (10) (2004) A1584–A1591.

[32] G. Koutitas, L. Tassiulas, A Delay Based Optimization Scheme for Peak Load Reduction in the Smart Grid, in: Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012, pp. 1–4.

[33] V. Rastogi, S. Nath, Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption, in: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), 2010, pp. 735–746.

[34] S. R. Rajagopalan, L. Sankar, S. Mohr, H. V. Poor, Smart Meter Privacy: A Utility-Privacy Tradeoff Framework, in: Proceedings of the 2nd IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011, pp. 190–195.

[35] Plugwise B.V., Smart Wireless Solutions for Energy Saving, Energy Monitoring and Switching, Available online: `http://www.plugwise.com` (2010).

[36] D. Comaniciu, P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.

[37] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, P. Armstrong, Power Signature Analysis, IEEE Power and Energy Magazine 1 (2) (2003) 56–63.

[38] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The WEKA Data Mining Software: An Update, SIGKDD Exploration Newsletter 11 (1) (2009) 10–18.